

УДК 004.047

Р. В. Олийнык

Украинская академия печати

**ПРОЕКТИРОВАНИЕ СТРУКТУРЫ GIT
ДЛЯ СОВМЕСТНОЙ ПОДГОТОВКИ ИЗДАНИЯ К ПЕЧАТИ**

Развитие современных информационных технологий, в частности полиграфии, представляет собой сложную систему, которая совершенствуется за счет качественно новых компонентов и услуг. Программное обеспечение, которое реализует концепции электронного документооборота, требует особого отношения к наличию и качеству сетевых инфраструктур, ведь информация должна быть предоставлена по первой необходимости в актуальном состоянии. Поэтому в сложившихся условиях все больше полиграфических предприятий используют облачные инфраструктуры, чтобы абстрагироваться от конкретного сервера хранения данных, а также обеспечить возможность параллельной работы в многопользовательском режиме над объектом издания, что позволяет сократить время подготовки издания к печати. Кроме того, многопользовательская среда требует не только обмена непосредственно объектом подготовки, но и резервного копирования с распределением зон ответственности между работниками. Поэтому для обеспечения качественного распределенного рабочего процесса возникает актуальная потребность использования систем контроля версий, которые позволяют частично или полностью автоматизировать процесс контроля версий готовящегося к печати издания, а также помогают предоставить всем участникам сети актуальную информацию о выбранной зоне ответственности издания по требованию. В результате интеграции систем контроля версий в многопользовательскую среду полиграфического предприятия можно минимизировать потери информации и увеличить прозрачность процесса обмена информацией.

Ключевые слова: контроль версий, многопользовательность, поток данных, производственный процесс, сетевая инфраструктура.

R. V. Oliynyk

Ukrainian Academy of Printing

DESIGN GIT STRUCTURE FOR JOINT PREPARATION FOR PRINTING

The development of modern information technologies, in particular polygraphy, is a complex system that is being improved through qualitatively new components and services. The software, which implements the concept of electronic document management, requires a special attitude regarding the availability and quality of network infrastructures, since information should be provided at the first necessity in the current state. Therefore, in the context of the growing number of printing companies, cloud-based environments are used to abstract from a particular data storage server, as well as to enable parallel work on the publication object in order to shorten the time it takes for the publication to be printed. In addition, the multiuser environment requires both the direct exchange of the object of preparation and the creation of backups and the allocation of areas of responsibility between employees. Therefore, in order to ensure a quality distributed workflow, there is an urgent need for using version control systems that allow partially or fully automate the process of controlling versions of the edited publication, as well as providing all participants with relevant information in the area of responsibility for publication on demand. As a result of the use of version control systems in the multi-authoritative environment, it will minimize the loss of information and increase the transparency of the information exchange process.

Key words: version control, multiuser, data flow, production process, network infrastructure.

Введение. На современном этапе развития полиграфической отрасли нередко осуществление допечатной подготовки происходит с привлечением разного количества специалистов узкого профиля, осуществляющих коллективную подготовку издания к печати. Для таких целей необходимы как средства резервного копирования исходных файлов каждого оператора, так и средства быстрого обмена изменениями в структуре документа с одновременным информированием и отправкой измененного объекта издания. Одним из методов обеспече-

ния коллективной работы над проектом является использование систем контроля версий, позволяющих следить за используемыми информационными потоками.

Основная часть. Системы контроля (управления) версий (Version Control System) представляют собой программное обеспечение, предназначенное для облегчения обмена информацией в процессе подготовки издания [1]. Такие средства, как система контроля версий (СКВ), предоставляют возможность сохранять несколько вариантов одного и того же доку-

мента и позволяют при необходимости вернуться к старым версиям, поэтому здесь можно узнать, кем были сделаны те или иные изменения. Такие средства коллективной разработки в большинстве используются при создании программного обеспечения, чтобы можно было сохранять исходные коды программ, однако сегодня наблюдается постепенная миграция и в другие области человеческой деятельности, в частности в полиграфию. СКВ позволяет разработчикам хранить предыдущие версии файлов и извлекать их оттуда. Она сохраняет информацию о версии каждого файла (и полную структуру проекта) в коллекции, которую часто называют репозиторием [2].

Выполненный анализ показал, что сегодня все системы контроля версий условно можно разделить на локальные и сетевые (централизованные и децентрализованные). Локальные с точки зрения коллективной работы являются неоптимальными, ведь здесь нет возможности быстрого обмена информацией между участниками сетевой инфраструктуры полиграфического производства, поэтому каждый оператор является изолированным в своей персонализированной области доступа. Сетевая СКВ устраняет недостаток обмена информацией, поскольку все рабочие документы хранятся децентрализованно в облачной среде и любой оператор, имея соответствующие уровни доступа, может воспользоваться не только исходными документами, но и просмотреть историю изменений файловой структуры. Такой подход позволяет осуществлять качественную коллективную подготовку издания к печати.

Выполненный анализ показал, что использование коллективной подготовки издания в полиграфии имеет место только в том случае, если вычислительная сетевая инфраструктура или средства пассивной передачи информации от оператора к оператору позволяют сохранять резервные копии, обеспечивая при этом режим быстрого возврата к последней локальной копии будущего издания. Таким образом, современное информационно-ориентированное полиграфическое предприятие получает возможность создавать совместное виртуальное рабочее пространство для хранения, копирования и архивирования данных, а также среду доступа к единым версиям локального программного обеспечения. В свою очередь, средства коллективной подготовки издания можно рассматривать как определенный набор правил доступа и обмена копиями рабочих файлов [3], ведь доступ к информации происходит в соответствии с определенными правами профиля пользователя. Поэтому для обеспечения качественной коллективной подготовки издания в полигра-

фии возникает актуальная потребность распределения издания по стадиям подготовки с одновременной интеграцией систем контроля версий, для чего необходимо исследовать структуру связей автор – подготовка – издание.

Как известно, процесс многопользовательской подготовки к печати требует участия определенного массива исполнителей, которые условно можно разделить по уровням иерархии. Здесь каждый уровень объединяет в себе группу исполнителей по должностным и функциональным обязанностям, которые имеют свои предметные области. На первом уровне иерархии находятся ответственные за выпуск лица, например главный редактор, главный дизайнер, менеджер издания. В должностные обязанности этой группы входит утверждение оригинала-макета, дизайна, решения конфликтных ситуаций в зонах ответственности авторов. Как правило, здесь работа ведется с конечным объектом или этапом издания. Следующая группа — это звено исполнителей, которые, находясь в своих зонах ответственности, проводят сегрегацию и оформление в соответствии с требованиями входных данных от внешних авторов. Последней группой можно считать лиц, занимающихся сбором общедоступных сведений, которые необходимы для подготовки издания к печати [3]. Организовав соответствующим образом подготовку издания и применив систему контроля версий, можно получить существенное преимущество. Поскольку сохраняются все промежуточные результаты в облаке, а доступ к последним копиям информации почти мгновенный, осуществляется загрузка только измененной информации, а не всех файлов в целом, но для этого необходимо провести привязку групп пользователей к СКВ.

Как показали проведенные исследования, интеграцию многопользовательской среды подготовки издания удобно осуществлять с системой контроля версий Git, поскольку это распределенная среда с поддержкой репозитория и системы ветвей (см. рисунок).

Главной ветвью, согласно с особенностями функционирования технологии Git, является Master-ветка. Здесь размещаются все готовые и проверенные элементы каждой отдельно взятой стадии подготовки издания, которые условно можно разделить на стадию подготовки, промежуточную и окончательные стадии. На *стадии подготовки* между всеми авторами разделяются зоны ответственности. Также на этой стадии в соответствии с правилами функционирования Git нужно решить, кто из операторов ветви *Master* многопользовательской системы будет иметь право собирать проект.



Интеграция Git и многопользовательской среды для подготовки издания к печати

Как правило, таким оператором является главный редактор или ответственный секретарь, поскольку именно он определяет окончательный макет будущего издания. Нужно заметить, что каждый автор многопользовательской среды может иметь свои Master-ветви, которые сливаются в одну и формируют собой целостный макет создаваемого издания. Все остальные операторы многопользовательской системы используют *Develop*-ветвь, в которой содержится вся исходная и рабочая информация, а также части объекта издания. Здесь размещен кроме текстового еще и иллюстративный материал. Работая над персонализированной рабочей ветвью *Develop* или *Master*, автор может создавать и другие ветви, например ветвь *Feature*, которая предназначена для нового объекта, независимого от других частей области доступа пользователя, но после слива в *Develop* становится одним целым с изданием.

В *промежуточной стадии* проходит процесс непосредственного взаимодействия зон ответственности всех авторов, поэтому есть риск возникновения конфликтов. Конфликт может произойти только тогда, когда несколько операторов многопользовательской системы изменяют одну и ту же зону ответственности, например, два корректора стилистически отредактируют одну и ту же часть текста дважды. В соответствии с правилами Git все конфликты разрешаются только в ручном режиме и никак иначе, тогда необходимо прийти к решению изменения или замены исполняемых файлов. Также промежуточная стадия подготовки изда-

ния предусматривает решение конфликтов в виде определенных *Hotfix* и добавление нового неоговоренного наполнения. Для таких целей удобно использовать ветку *Feature*, которая в случае необходимости позволит осуществить возврат к оригинальному макету издания без потери. Ветви *Hotfix* создаются при необходимости внесения исправления в уже готовый макет [2]. Позже все ветви каждого отдельно взятого автора и его персонализированной области доступа сливаются в ветку *Release* и загружаются на облачный сервер, откуда все остальные участники многопользовательской публикации могут получить копию вместе со всеми актуальными изменениями на момент слияния.

Последней стадией в подготовке издания считают *окончательную стадию*. В случае отсутствия ошибок на этой стадии предполагаются процессы слияния всех ветвей в единую ветку *Release* и формирования исходных данных объекта издания для добавления полученного результата к системе управления рабочими потоками. Поэтому, исправив все ошибки и осуществив объединение всех ветвей *Release*, в *Master* получается завершенная публикация, которую можно отдавать в печать.

Как известно, в современной полиграфии все большее количество производственных процессов совершаются с помощью технологии управления потоками данных CIP4. Главная идея CIP4 — это сбор актуальных сведений о состоянии технологических процессов и их передача на различные звенья производства в целях управления ими.

Исследование децентрализованных систем контроля версий, в том числе в полиграфической отрасли, показало отсутствие стандартных средств для управления рабочими потоками данных [1], но для решения этой проблемы можно использовать механизм модульности. Поскольку исследуемая облачная инфраструктура должна обеспечивать гибкое управление рабочими потоками данных, необходимо реализовать расширение формата данных JDF/JMF за счет введения новых тегов спецификации, которые будут созданы с помощью парсинга коммита Git, что позволит просматривать и вносить изменения в файловую структуру данных XML в автоматическом режиме [1].

Такая частичная интеграция удобна прежде всего тем, что две системы, CIP4 и Git пересекаются весьма посредственно, поскольку в Git сохраняются все снимки готовящегося объекта издания, представляющие собой с точки зрения CIP4 лишнюю, ненужную информацию, ведь здесь критическим параметром является корректность и скорость потоков данных, поскольку, как известно, CIP4 не несет в себе исполнительный файл, а только содержит адрес, где размещен нужный документ [1]. Поэтому в соответствии с потребностями протокола CIP4 вместе с подготовленным макетом необходимо передать название, место расположения репо-

зитория издания, а также информацию об авторе. Здесь автором выступает главный редактор или любой, кто проводит генерирование ветви Release и определяет местоположение дополнительных составных частей [1]. Всю же остальную информацию целесообразно в дальнейшем сохранять в репозитории Git. Таким образом, интеграция Git в CIP4 выглядит как интеграция дополнительного модульного решения, которое может использоваться на предприятиях, где есть многопользовательская система подготовки макета к публикации.

Заключение. Предложенная модель интеграции Git в многопользовательскую среду и систему управления рабочими потоками CIP4 позволит осуществлять полностью независимую работу над проектом несколькими авторам в распределенных персонализированных областях доступа, обмениваясь при этом всеми частями издания в режиме реального времени с помощью Git. К тому же нивелируется возможность потери информации, ведь Git всегда сохраняет историю всех изменений. Кроме того, вместе с модификацией рабочей области осуществляется обмен файлами, используемыми при подготовке издания, а интеграция информации о репозиториях Git в CIP4 позволяет вернуться к подготовке, загрузив не только конечный вариант издания, но и всю историю.

Литература

1. Олійник Р. В. Модель структур данных рабочих потоков JDF в облачной инфраструктуре // Моделирование и информационные технологии: сб. науч. тр. Нац. акад. наук Украины. Киев, 2014. № 72. С. 59–63.
2. Чакон С., Штрауб Б. Git для профессионального программиста [Электронный ресурс]. СПб.: Питер, 2016. 496 с. (pdf).
3. Oliynyk R. Optimization of multi-architecture network based on cloud computing // Technical sciences: modern issues and development prospects: International Conference. Sheffield, UK, 2013. P. 75–76.

References

1. Oliynyk R. V. Model structure of these working flows of JDF in the cloud infrastructure. *Modelirovaniye i informatsionnyye tekhnologii: sbornik nauchnykh trudov Natsional'noy akademii nauk Ukrainy* [Simulation and informational technologies: collection of scientific papers by the National Academy of Sciences of Ukraine], Kiev, 2014, no. 72, pp. 59–63 (In Ukrainian).
2. Chacon S., Straub B. Git dlya professional'nogo programmista [Git for a professional programmer] [Electronic resource]. St. Petersburg, Piter Publ., 2016. 496 p. (pdf).
3. Oliynyk R. Optimization of multi-architecture network based on cloud computing. *International Conference "Technical sciences: modern issues and development prospects"*, Sheffield, UK, 2013. P. 75–76.

Информация об авторах

Олійник Роман Владимирович – кандидат технических наук, ассистент кафедры автоматизации и компьютерных технологий. Украинская академия печати (79020, г. Львов, ул. Пид Голоском, 19, Украина). E-mail: enigmus@ukr.net

Information about the authors

Oliynyk Roman Vladimirovich – PhD (Engineering), assistant lecturer, the Department of Automation and Computer Technologies. Ukrainian Academy of Printing (19, Pid Goloskom str., 79020, Lvov, Ukraine). E-mail: enigmus@ukr.net

Поступила 14.08.2017